

Peer-Review 2: Network

Emanuele Puricelli, Salvatore Scarfone, Luca Proserpio

Gruppo 50

Valutazione del diagramma UML della Network del gruppo 60.

Lati positivi

Utilizzare il valore del token per vedere se il giocatore che ha mandato il messaggio è effettivamente il giocatore del turno e controllare il successo della comunicazione è una soluzione semplice e funzionale per la gestione delle comunicazioni con i vari client.
Utilizzare un ServerThread per ogni Client rende semplice la loro gestione.

Lati negativi

Avere direttamente il Server che comunica con il controller non dà la possibilità di avere più partite che girano sulla stessa macchina al meno che non vengano lanciati più server con porte diverse, che li renderebbe più difficili da gestire.

Controllare il token direttamente sul Server invece che su una classe intermedia può rendere difficile l'aggiornamento della View dei giocatori che non stanno eseguendo il turno, in quanto il server stesso dovrebbe gestire direttamente quali informazioni inviare o tenere nascoste ai vari client per ogni modifica della View.

Confronto tra le architetture

I punti di forza è principalmente l'utilizzo dei token, la nostra architettura utilizza scelte abbastanza simili per controllare chi è il giocatore che manda i messaggi ma non abbiamo un modo simile per informare il Server della ricezione, cosa che potremmo includere in modo simile così da non dovere mandare messaggi di ricezione avvenuta al Server.